

Міністерство освіти і науки, молоді та спорту України
Державний вищий навчальний заклад
"Національний гірничий університет"

Кафедра систем електропостачання

Методичні вказівки
до виконання лабораторної роботи МП-6
„Вивчення емулятора роботи мікропроцесора KP580VM80”
для студентів, які навчаються за напрямами підготовки:
6.050701 „Електротехніка та електроенергетика ”,
6.050702 „Електромеханіка ”

Дніпропетровськ
2012

Методичні вказівки до виконання лабораторної роботи МП-6 "Вивчення емулятора роботи мікропроцесора КР580ВМ80" для студентів, які навчаються за напрямами підготовки: 6.050701 "Електротехніка та електроенергетика", 6.050702 "Електромеханіка" / Укл.: Г.М.Бажін, А.В.Рухлов, С.В.Дибрін, Д.О.Кошовий – Дніпропетровськ: Державний ВНЗ "НГУ", кафедра систем електропостачання, 2012. - 15 с.

1. Мета роботи

1.1. Вивчити структуру, призначення, функціональні можливості та принцип шинної організації мікропроцесорної системи.

1.2. Вивчити склад, структурну організацію і призначення основних екранних елементів емулятора роботи мікропроцесора KP580BM80.

1.3. Вивчити методи роботи й опанувати навички діалогової взаємодії користувача з емулятором роботи мікропроцесора KP580BM80.

2. Теоретичні положення роботи

Емулятор роботи мікропроцесора KP580BM80 розроблено на кафедрі обчислювальної техніки Московського інституту радіотехніки, електроніки та автоматики (МІРЕА) і призначено для вивчення функціонування мікропроцесора у складі мікропроцесорної системи й отримання практичних навичок програмування його роботи.

До складу мікропроцесорної системи (МПС) входять (*рис 1*): мікропроцесор (МП), генератор тактових імпульсів (ГТІ), постійний запам'ятовуючий пристрій (ПЗП), оперативний запам'ятовуючий пристрій (ОЗП), інтерфейси введення-виведення (ІВВ), шина адреси (ША), шина даних (ШД) і шина управління (ШУ).

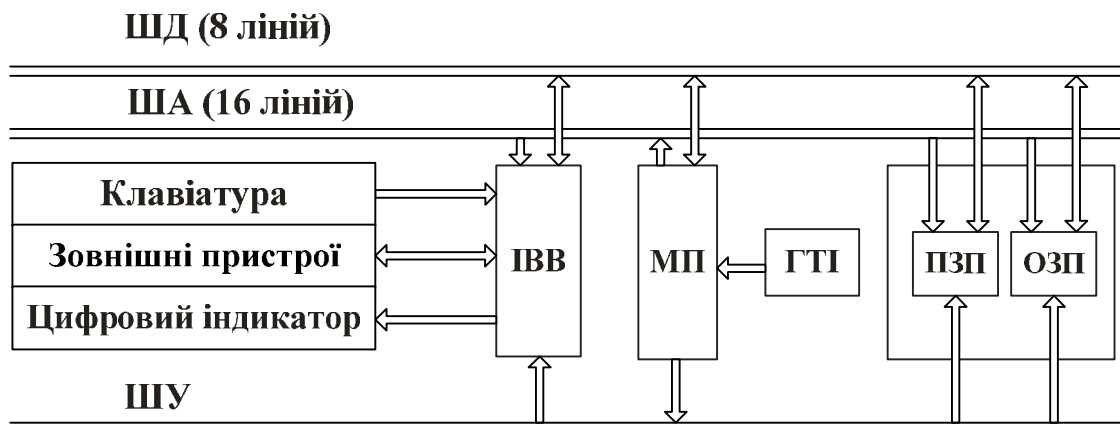


Рис.1. Архітектура мікропроцесорної системи на основі мікропроцесора KP580BM80

Мікропроцесор виконує обробку інформації, генератор тактових імпульсів задає темп роботи системи. Оперативний запам'ятовуючий пристрій служить для тимчасового зберігання програм і даних, що використовуються при обробці інформації. При відключенні системи від джерела живлення інформація, розміщена в ОЗП, втрачається. Постійний запам'ятовуючий пристрій здійснює зберігання програм довготривалої дії і характеризується тим, що при вимкненні живлення інформація в ньому не знищується. Інтерфейси введення-виведення

призначені для здійснення зв'язку з зовнішніми пристроями. Шини – це лінії зв'язку між елементами системи. Шина адреси односпрямована і нею мікропроцесор здійснює вибір адрес пам'яті й інтерфейсів введення-виведення. Шина даних призначена для обміну даними між МП, пам'яттю й інтерфейсами. Шиною управління МП здійснює управління елементами системи.

У мікропроцесорній системі діалоговий режим організовується за допомогою клавіатури (або її імітатора) і системної програми, що має назву Монітор. Монітор розміщується в постійному запам'ятовуючому пристрої, тобто є резидентною програмою. Іншими словами, **Монітором** називається резидентна системна програма діалогової взаємодії оператора з МПС. Монітори бувають різної складності й об'єму – залежно від режиму роботи, оснащення МПС периферійним устаткуванням, числа оброблюваних команд і т.п. У системі, що вивчається, роль програми Монітор виконує емулятор. До його функцій входять:

- управління розміщенням прикладної програми в пам'яті МПС і її наладкою;
- управління режимами роботи МПС;
- управління виконанням прикладної програми.

Робота користувача МПС зводиться до занесення програми в ОЗП, її запуску, налагодження й отримання результатів виконання.

Робота з емулятором починається із запуску програми CPU580, розташованої на робочому столі. Вікно, що відкривається на екрані (*рис 2*), містить ряд екранних полів, кнопок та екранних покажчиків. У верхній частині екрану розміщено поля: АДРЕС, ДАННЫЕ, СОСТОЯНИЕ, ПОЗИЦИЯ і ФЛАГИ. Їх призначення буде розглянуто пізніше. Ліворуч на екрані розміщено два чорних поля зі знаками питання. Вони призначені для перегляду і зміни вмісту регістрів та комірок пам'яті. Внесення змін здійснюється за допомогою цифрової клавіатури, розміщеної під чорними полями. Вибір об'єкту доступу здійснюється за допомогою клавіатури доступу і функціональних клавіш праворуч від цифрової клавіатури. Вміст регістрів МП у даний момент показано в полі РЕГИСТРЫ. Вміст комірок області пам'яті можна проглянути в білому полі праворуч на екрані. Воно має вигляд таблиці зі стовпцями АДРЕС, КОД, КОМАНДА і називається полем адрес і програм.

До складу **цифрової клавіатури** входять:

- 1) клавіші 0...F – для введення 16-ричних чисел від 0 до F;
- 2) клавіша ЗАБОЙ – відмінняє помилкове введення інформації;
- 3) клавіша ВВОД – дозволяє запис набраної інформації до регістрів МП або комірки пам'яті.

Клавіатура доступу до регістрів та комірок пам'яті містить клавіші наступного призначення:

- 1) клавіша ОЗУ – дозволяє заповнення або зміну вмісту комірки пам'яті;

- 2) клавіша РЕГИСТР – дозволяє заповнення або зміну вмісту регістрів МП;
- 3) клавіша РсН – викликає на дисплей вміст старшого байта лічильника команд МП для подальшої його модифікації;
- 4) клавіша РсL – викликає на дисплей вміст молодшого байта лічильника команд МП для подальшої його модифікації;
- 5) клавіша SpН – викликає на дисплей вміст старшого байта покажчика стека МП для подальшої його модифікації;
- 6) клавіша SpL – викликає на дисплей вміст молодшого байта покажчика стека МП для подальшої його модифікації.

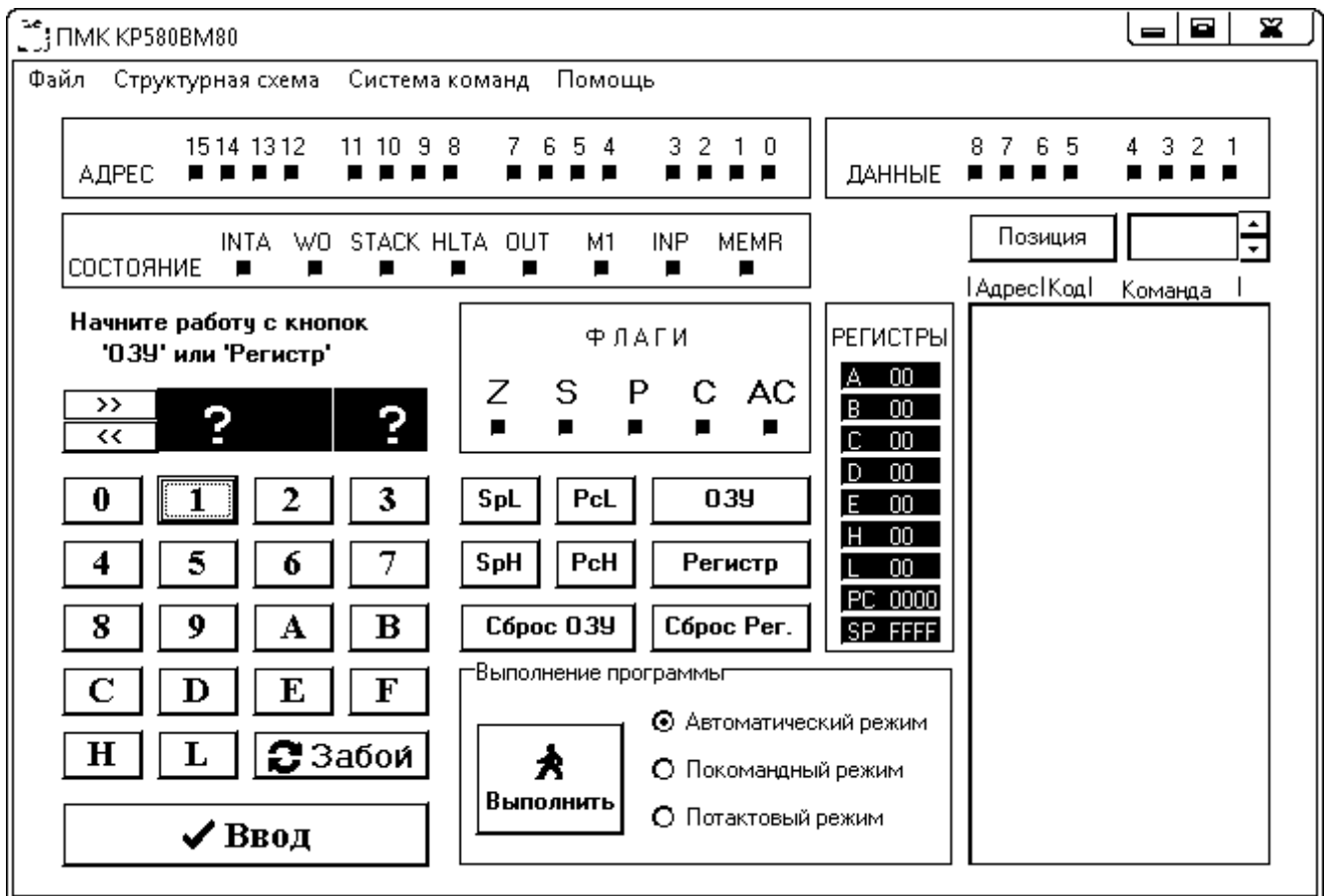


Рис.2. Вікно емулятора роботи мікропроцесора KP580BM80 розробленого на кафедрі обчислювальної техніки МІРЕА

До функціональних клавіш відносяться :

- 1) клавіша СБРОС ОЗУ – призначена для обнуління вмісту всього простору пам'яті МП;
- 2) клавіша СБРОС РЕГ. – потрібна для обнуління вмісту всіх регістрів і флагів МП.

Перемикачі АВТОМАТИЧЕСКИЙ РЕЖИМ, ПОКОМАНДНЫЙ РЕЖИМ, ПОТАКТОВЫЙ РЕЖИМ призначені для включення одного з трьох режимів виконання програм: автоматично – після запуску виконується вся програма; покомандно – виконання програми здійснюється кроками по одній команді; потактово – кроками по одному такту (кожна команда містить декілька тактів).

Клавіша **ВЫПОЛНИТЬ** призначена для виконання програм користувача у вибраному режимі.

Перегляд і зміна вмісту реєстрів МП виконується в такій послідовності:

1) натиснути клавішу **РЕГИСТР** і вибрати клавішу з позначенням необхідного реєстра – емулятор готовий до введення інформації;

2) якщо необхідно проглянути декілька реєстрів, то натисканням клавіші >> можна вивести на дисплей їх вміст в наступній послідовності: А, В, С, D, E, F, H, L, Z, W. Реєстри Z і W не модифікуються, можливий тільки їх перегляд. Клавіша << дозволяє переглядати реєстри в протилежному напрямі, при цьому ліворуч на дисплеї відображується ім'я обраного реєстра, а праворуч – його вміст;

3) зміна вмісту реєстра виконується набором на цифровій клавіатурі необхідного нового вмісту, після чого натискається клавіша **ВВОД**.

Перегляд і зміна вмісту комірок пам'яті виконується в наступній послідовності:

1) натиснути клавішу **ОЗУ** і набрати на цифровій клавіатурі необхідну адресу, яка відображується при цьому ліворуч на чорному полі дисплея. Праворуч на чорному полі відображується вміст комірки пам'яті з вибраною адресою. Одночасно в білому полі правої частини екрану з'являються адреси і вміст комірок області пам'яті;

2) при необхідності перегляду підряд декількох комірок пам'яті, натисканням клавіші >> можна виводити на дисплей їх адреси послідовно у бік збільшення з індикацією вмісту і можливістю внесення змін. Клавіша << дозволяє здійснювати ту ж операцію в протилежному напрямі;

3) змінити вміст комірки пам'яті, тобто записати до неї нове значення, можна, набираючи його на цифровій клавіатурі з наступним натисканням клавіші **ВВОД**. При цьому на чорному полі відбувається автоматичний перехід до наступної адреси і відображується її вміст. Змінений вміст буде показаний в полі адрес і програм праворуч на екрані.

Процес і результати виконання програми можна переглядати також за структурною схемою мікропроцесора, яку можна викликати на екран, вибравши в головному меню (верхній рядок на екрані) позицію **СТРУКТУРНАЯ СХЕМА** (рис. 3). Ця схема є архітектурою мікропроцесора з указанням реєстрів, їх вмісту і взаємозв'язків між ними. Окрім зображення структурної схеми на екранній сторінці продубльовано всі клавіші основної сторінки, окрім **PcH, PcL, SpH, SpL**

і ПОЗИЦІЯ. Повернення до початкової сторінки здійснюється натисненням кнопки Выход.

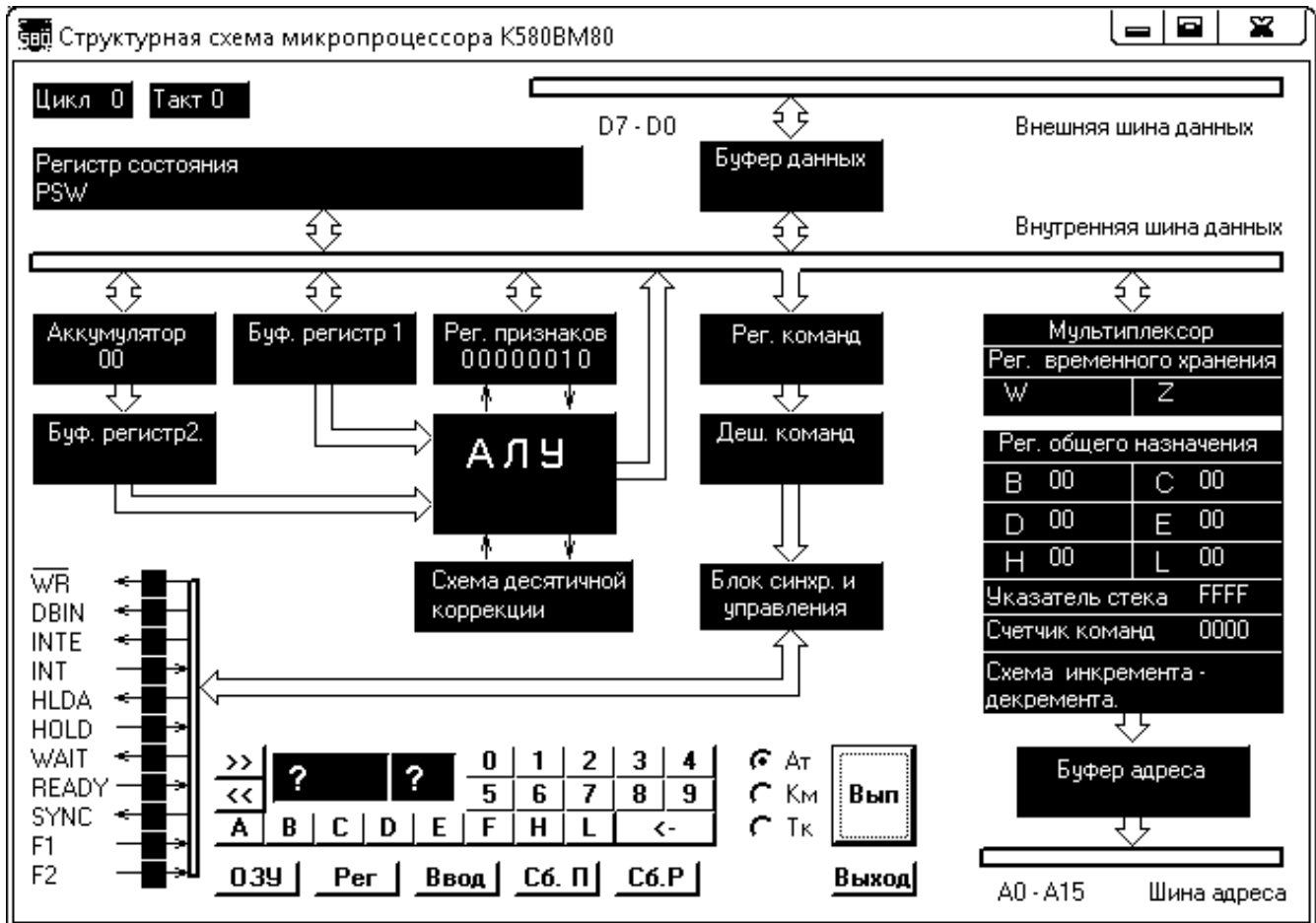


Рис.3. Вікно „Структурна схема мікропроцесора KP580BM80” емулятора роботи мікропроцесора KP580BM80

Система команд мікропроцесора і короткі пояснення роботи з емулятором можуть бути викликані на екран через головне меню (позиції СИСТЕМА КОМАНД і ПОМОЩЬ). Для зручності користування система команд мікропроцесора представлена також в додатку (таблиця П1) до цієї роботи.

3. Практичні рекомендації та приклади виконання роботи

Для освоєння методів перегляду і зміни вмісту регістрів та комірок пам'яті, а також отримання первинних навичок роботи з емулятором мікропроцесора розглянемо наступний приклад.

Приклад: Програма складання двох шістнадцятиричних чисел 2FH і 4EH і запам'ятовування суми в ОЗП на мові асемблера має вигляд:

MVI, 2F – розміщення в регістрі А числа 2F;

MVI C, 4E – розміщення в регістрі С числа 4E;

ADD C – складання вмісту регістра С з вмістом регістра А, результат – в регістрі А;

STA 23FE – пересилка результату з регістра А до комірки пам'яті з адресою 23FEH.

Необхідно занести цю програму до ОЗП, запустити її й отримати результат. Запуск програми виконати в автоматичному режимі.

За допомогою таблиці П1 з додатків або на екранній сторінці СИСТЕМА КОМАНД знайдемо коди команд асемблера (код команди складається з двох цифр у шістнадцятиричній системі числення, що визначають положення відповідної команди в таблиці кодів команд). Розмістимо програму в ОЗП, починаючи з адреси 2150H. Програма набере вигляду, представленого в табл. 1.

Таблиця 1

Адреса комірки пам'яті	Код команди	Мнемоніка (команда на мові асемблера)	Коментарі
2150 2151	3E 2F	MVI A, 2F	Завантаження до акумулятора числа 2FH
2152 2153	0E 4E	MVI C, 4E	Завантаження до регістру С числа 4EH
2154	81	ADD C	Складання вмісту регістрів А і С. Результат – в акумуляторі
2155 2156 2157	32 FE 23	STA 23FE	Пересилання результату до комірки пам'яті з адресою 23FEH
2158	76	HLT	Зупинка

Занесення програми до ОЗП емулятора проводиться в такій послідовності:

- натиснути клавішу ОЗУ (вибір режиму зміни вмісту комірок пам'яті);
- набрати 2150 цифровими клавішами (адреса початку програми);
- набрати 3E цифровими клавішами (запис до комірки коду 3EH);
- натиснути клавішу ВВОД (запам'ятовування цього коду і перехід до наступної комірки 2151H);
- набрати 2F цифровими клавішами (запис до комірки 2151 коду 2FH);
- натиснути клавішу ВВОД (запам'ятовування цього коду і перехід до

наступної комірки 2152H);

... і так далі (згідно з табл. 1);

– набрати 76 цифровими клавішами – запис коду зупинки до комірки 2158;

– натиснути клавішу ВВОД (запам'ятовування цього коду).

Внаслідок цих дій в ОЗП буде записано задану програму.

Запуск програми здійснюється в такій послідовності.

1. У лічильнику команд встановлюється початкова адреса програми, для чого:

– натиснути клавішу СБРОС РЕГ. (обнуління вмісту регістрів);

– натиснути клавішу РЕГИСТР (вибір режиму заповнення регістрів);

– натиснути клавішу РСН (вибір старшого байта лічильника команд);

– набрати 21 (старший байт адреси);

– натиснути клавішу ВВОД (запис до лічильника команд старшого байта адреси);

– натиснути клавішу РСЛ (вибір молодшого байта лічильника команд);

– набрати 50 (молодший байт адреси);

– натиснути клавішу ВВОД (запис до лічильника команд молодшого байта адреси).

В результаті цих дій до лічильника команд буде записано адресу початку програми (2150H).

2. Здійснюється вибір позиції:

– натиснути клавішу ОЗУ;

– набрати 2150;

– натиснути клавішу ПОЗИЦИЯ.

В результаті цих дій буде вибрано область пам'яті, а в полі адрес і програм буде представлено початкову програму, записану на мові асемблера.

3. Вибирається режим виконання програми – установкою перемикача АВТОМАТИЧЕСКИЙ РЕЖИМ в полі ВЫПОЛНЕНИЕ ПРОГРАММЫ.

4. Запуск програми виконується натисненням клавіші ВЫПОЛНИТЬ.

В результаті виконання програми в регістр А буде поміщено шукану суму (7D), в регістрі С залишиться число (4E), записане в процесі виконання програми, в лічильнику команд – адреса комірки пам'яті, наступної після програми (2159H).

У полях АДРЕС і ДАННЫЕ червоним будуть вказані розряди шини адреси і даних, що були задіяні в останньому циклі виконання програми. У двійковій формі це – адреса 0010 0001 0101 1001 та дані 0111 0110. У 16-ричній формі це – адреса 2159H (адреса в лічильнику команд після виконання програми) і число 76H – код останньої команди програми.

5. Порядок виконання роботи

1. За конспектом лекцій вивчити: структуру, призначення, функціональні можливості та принцип шинної організації МПС. Як наслідок необхідно знати: призначення мікроконтролера; структуру мікропроцесорної системи, її шинну організацію; призначення МП, ОЗП, ПЗП, інтерфейсів введення-виводу; призначення шин адреси, даних і управління, їх розрядність.

2. Вивчити склад, структурну організацію і призначення основних екранних елементів емулятора роботи мікропроцесора KP580BM80.

3. Розібрати наведений приклад складання і асемблювання програми.

4. Відповідно до індивідуального завдання (табл. П2) скласти програму його виконання. Номер варіанту визначається номером студента в списку групи. Користуючись таблицею кодів команд (табл. П1), виконати асемблювання програми свого варіанту.

5. Виконати вручну розрахунки завдання в двоїчній формі. Результат перевести у 16-ричну форму.

6. Занести складену програму в ОЗП емулятора з заданою початковою адресою, запустити її й отримати результат.

7. Показати результати виконання програми і зроблені розрахунки викладачеві, отримати його підпис, який свідчить про виконання завдання.

Завдання вважається виконаним, якщо результат розрахунків і результат виконання програми на емуляторі співпадають.

6. Зміст звіту

Звіт повинен містити:

- номер лабораторної роботи;
- найменування роботи;
- мету роботи;
- проасембльоване завдання свого варіанту, оформлене у вигляді таблиці;
- розрахунки завдання у двоїчній формі та результат у 16-ричній формі.

7. Контрольні питання

1. У чому полягає принцип шинної організації мікропроцесорної системи?
2. Яке призначення ОЗП, ПЗП, МП та інтерфейсу введення-виводу?
3. Яка розрядність шин адреси і даних мікропроцесора K580VB80?
4. Яка з шин мікропроцесора є односпрямованою?
5. Дати поняття і вказати призначення програми Монітор.
6. Проасемблювати запропоновану викладачем програму.
7. Як проводиться занесення програми до пам'яті МПС?
8. В якій системі числення вводяться дані до пам'яті МПС?
9. Як здійснюється запуск виконання програми?
10. Якою кількістю шістнадцятиричних символів може бути записане восьмирозрядне двоїчне число?
11. В якій пам'яті (ОЗП або ПЗП) зберігається Монітор?

8. Рекомендована література

Конспект лекцій по курсу „Микропроцессорная техника” для студентов направлений подготовки 6.050701 „Электротехника и электротехнологии” и 6.050702 „Электромеханика”. Часть 1 / Сост. Г.М. Бажин. – Днепропетровск: НГУ, каф СЭС, 2008. – 44 с.

Додатки

Таблиця III. Коди команд мікропроцесора K580BM80

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LXI B,&	STAX B	INX B	INR B	DCR B	MVI B,#	RLC	-	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,#	RRC
1	-	LXI D,&	STAX D	INX D	INR D	DCR D	MVI D,#	RAL	-	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E,#	RAR
2	-	LXI H,&	SHLD *	INX H	INR H	DCR H	MVI H,#	DAA	-	DAD H	LHLD *	DCX H	INR L	DCR L	MVI L,#	CMA
3	-	LXI SP&	STA *	INX SP	INR M	DCR M	MVI M,#	STC	-	DAD SP	LDA *	DCX SP	INR A	DCR A	MVI A,#	CMC
4	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MOV C,M	MOV C,A
5	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MOV E,M	MOV E,A
6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MOV L,M	MOV L,A
7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MOV A,M	MOV A,A
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
B	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
C	RNZ	POP B	JNZ *	JMP *	CNZ *	PUSH B	ADI #	RST O	RZ	RET	JZ *	-	CZ *	CAL L	ACI #	RST 1
D	RNC	POP D	JNC *	OUT N	CNC *	PUSH D	SUI #	RST 2	RS	-	JC *	IN N	CC *	* -	SBI *	RST 3
E	RPO	POP H	JPO *	XTH L	CPO *	PUSH H	ANI #	RST 4	RPE	PCHL	JPE *	XCHG	CPE *	-	XRI #	RST 5
F	RP	POP PSW	JP *	DI	CP *	PUSH PSW	ORI #	RST 6	RM	SPHL	JM *	EI	CM *	-	CPI #	RST 7
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Примітки:

N - номер порту введення-виводу;

& – двобайтовий операнд D16;

* – двобайтовий операнд adr;

– однобайтовий операнд D8;

PSW - вміст акумулятора і регістра ознак (слово стану процесора).

ПРИКЛАД: код операції – C3; операція – JMP adr.

Таблиця П2. Варіанти індивідуальних завдань

Номер варіанту	Програма на мові асемблера	Початкова адреса	Коментар до програми
1	MVI C, 2B MVI A, 32 ADD C STA 2300	2100	Занесення до регістру С числа 2ВН Занесення до регістру А числа 32Н Складання вмісту регістрів А і С Пересилка результату до комірки пам'яті з адресою 2300Н
2	MVI B, 31 MVI A, 38 SUB B STA 2310	2110	Занесення до регістру В числа 31Н Занесення до регістру А числа 38Н Віднімання вмісту В від вмісту А Пересилка результату до комірки пам'яті з адресою 2310Н
3	MVI D, 5A MVI A, 52 ADD D STA 2320	2120	Занесення до регістру D числа 5АН Занесення до регістру А числа 52Н Складання вмісту регістрів А і D Пересилка результату до комірки пам'яті з адресою 2320Н
4	MVI E, F1 MVI A, 5A SUB E STA 2325	2130	Занесення до регістру Е числа F 1Н Занесення до регістру А числа 5АН Віднімання вмісту Е від вмісту А Пересилка результату до комірки пам'яті з адресою 2325Н
5	MVI H, 0E MVI A, 21 ADD H STA 232F	2140	Занесення і регістр Н числа 0ЕН Занесення в регістр А числа 21Н Складання вмісту регістрів А і Н Пересилка результату до комірки пам'яті з адресою 232FH
6	MVI L, A8 MVI A, 2F ADD L STA 232E	2150	Занесення до регістру L числа A8Н Занесення до регістру А числа 2FH Складання вмісту регістрів А і L Пересилка результату до комірки пам'яті з адресою 232ЕН
7	MVI B, 2B MVI A, 1C SUB B STA 231E	2160	Занесення до регістру В числа 2ВН Занесення до регістру А числа 1СН Віднімання вмісту В від вмісту А Пересилка результату до комірки пам'яті з адресою 231ЕН
8	MVI C, 1F MVI A, F1	2170	Занесення до регістру С числа 1FH Занесення в регістр А числа F1Н

Номер варіанту	Програма на мові асемблера	Початкова адреса	Коментар до програми
	SUB C STA 2317		Віднімання вмісту C від вмісту A Пересилка результату до комірки пам'яті з адресою 2317H
9	MVI D, 3E MVI A, FE SUB D STA 2322	2180	Занесення до регістру D числа 3EH Занесення до регістру A числа FEH Віднімання вмісту D від вмісту A Пересилка результату до комірки пам'яті з адресою 2322H
10	MVI H, 40 MVI A, 8F ADD H STA 230E	2190	Занесення до регістру H числа 40H Занесення до регістру A числа 8FH Складання вмісту регістрів A і H Пересилка результату до комірки пам'яті з адресою 230EH
11	MVI L, 28 MVI A, 7A ADD L STA 230F	21A0	Занесення до регістру L числа 28H Занесення до регістру A числа 7AH Складання вмісту регістрів A і L Пересилка результату до комірки пам'яті з адресою 230FH
12	MVI E, 5E MVI A, 5A ADD E STA 2301	21B0	Занесення до регістру E числа 5EH Занесення до регістру A числа 5AH Складання вмісту регістрів A і E Пересилка результату до комірки пам'яті з адресою 2301H
13	MVI B, 0F MVI A, FD SUB B STA 2304	21C0	Занесення до регістру B числа 0FH Занесення до регістру A числа FDH Віднімання вмісту B від вмісту A Пересилка результату до комірки пам'яті з адресою 2304H
14	MVI H, 1A MVI A, FF SUB H STA 231A	21D0	Занесення до регістру H числа 1AH Занесення до регістру A числа FEH Віднімання вмісту H від вмісту A Пересилка результату до комірки пам'яті з адресою 231AH
15	MVI D, 1B MVI A, FE SUB D STA 231B	21E0	Занесення до регістру D числа 1BH Занесення в регістр A числа FEH Віднімання вмісту D від вмісту A Пересилка результату до комірки пам'яті з адресою 231BH

Номер варіанту	Програма на мові асемблера	Початкова адреса	Коментар до програми
16	MVI C, 2C MVI A, CC SUB C STA 231C	21F0	Занесення до регістру C числа 2CH Занесення до регістру A числа CCH Віднімання вмісту C від вмісту A Пересилка результату до комірки пам'яті з адресою 231CH
17	MVI L, 3D MVI A, 8F SUB L STA 232B	2200	Занесення до регістру L числа 3DH Занесення до регістру A числа 8FH Віднімання вмісту L від вмісту A Пересилка результату до комірки пам'яті з адресою 232BH
18	MVI E, 4E MVI A, 71 SUB E STA 230A	2210	Занесення до регістру E числа 4EH Занесення до регістру A числа 71H Віднімання вмісту E від вмісту A Пересилка результату до комірки пам'яті з адресою 230AH
19	MVI A, 2F MVI B, 1E SUB B STA 2303	2220	Занесення до регістру A числа 2FH Занесення до регістру B числа 1EH Віднімання вмісту B від вмісту A Пересилка результату до комірки пам'яті з адресою 2303H
20	MVI A, FE MVI D, E4 SUB D STA 2304	2230	Занесення до регістру A числа FEH Занесення до регістру D числа E4H Віднімання вмісту D від вмісту A Пересилка результату до комірки пам'яті з адресою 2304H
21	MVI A, OE MVI H, EA ADD H STA 232A	2240	Занесення до регістру A числа OEH Занесення до регістру H числа EAH Складання вмісту регістрів A і H Пересилка результату до комірки пам'яті з адресою 232AH
22	MVI A, 08 MVI L, EF ADD L STA 232E	2250	Занесення до регістру A числа 08H Занесення до регістру L числа EFH Складання вмісту регістрів A і L Пересилка результату до комірки пам'яті з адресою 232EH
23	MVI A, 19 MVI C, 19 ADD C	2260	Занесення до регістру A числа 19H Занесення до регістру C числа 19H Складання вмісту регістрів A і C

Номер варіанту	Програма на мові асемблера	Початкова адреса	Коментар до програми
	STA 230D		Пересилка результату до комірки пам'яті з адресою 230DH
24	MVI A, 28 MVI E, 28 ADD E STA 2320	2270	Занесення до регістру А числа 28H Занесення до регістру Е числа 28H Складання вмісту регістрів А і Е Пересилка результату до комірки пам'яті з адресою 2320H
25	MVI A, F1 MVI E, 18 SUB E STA 2301	2280	Занесення до регістру А числа F1H Занесення до регістру Е числа 18H Віднімання вмісту Е від вмісту А Пересилка результату до комірки пам'яті з адресою 2301H